REMARKS

Claims 1-20 are now pending in the application. Claim 14 is amended. Support for the amendments can be found at paragraphs [0013]-[0018] of the originally filed specification. The Examiner is respectfully requested to reconsider and withdraw the rejection(s) in view of the amendments and remarks contained herein.

REJECTION UNDER 35 U.S.C. § 101

Claims 14-20 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. This rejection is respectfully traversed.

Claims 14 is amended herein to recite that the reference count is maintained in a computer readable medium. Therefore, the claimed subject matter achieves a tangible result, and, as a result, is directed to statutory subject matter.

Accordingly, Applicants respectfully request the Examiner reconsider and withdraw he rejection of claim 14 under 35 U.S.C. § 101, along with rejection on these grounds of all claims dependent therefrom.

REJECTION UNDER 35 U.S.C. § 112

Claims 2, 6, 9, and 14 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point and distinctly claim the subject matter which Applicant regards as the invention. This rejection is respectfully traversed.

Applicants have amended claims 2, 6, and 9 as suggested by the Examiner. However, Applicants conclude from the Examiner's remarks regarding claim 14 that the rejection of claim 14 under 35 U.S.C. §112, second paragraph, was misidentified. In particular, Applicants believe the Examiner intended to reject claim 15 instead of claim

14 under 35 U.S.C. §112, second paragraph. Therefore, Applicants have amended claim 15 according to the Examiner's suggestions for amending claim 14.

Accordingly, Applicants respectfully request the Examiner reconsider and withdraw the rejection of claims 2, 6, 9, and 14 under 35 U.S.C. § 112, second paragraph.

REJECTION UNDER 35 U.S.C. § 103

Claims 1-6, 8-10, and 12-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ma et al. (U.S. Pat. No. 5,920,725) in view of APA Admitted Prior Art. This rejection is respectfully traversed.

The Examiner relies on Ma et al. to teach that each software module maintains its own reference count. However, Ma et al. teach at col. 10, lines 40-56 that replaced objects must be marked invalid in order for new references to be made to the presumed valid replacement objects. But it is not conventional for a Kernel to direct requests for application functions in this way. Also, tracking object validity, a *variable* state, for each object is an inconvenient and error prone process. Therefore, it is significant that Ma et al. do not teach, suggest, or motivate maintaining a count for each loadable module as well as for each version of a loadable module, so that when the replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module based on version information registered with the kernel.

The Examiner merely relies on APA to teach that some execution environments, like Linux, allow replacement of a loadable software module by first removing the module. However, Ma et al. in view of APA do not teach, suggest, or motivate

maintaining a count for each loadable module as well as for each version of a loadable module, so that when the replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module based on version information registered with the kernel.

Applicants' claimed invention is directed toward loading a replacement software module without first having to unload a previous version of the software module. In particular, Applicant's claimed invention is directed toward maintaining a count for each loadable module as well as for each version of a loadable module, so that when the replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module based on version information registered with the kernel. For example, independent claim 1, especially as amended, recites, "maintaining a reference count for a given software module loaded in the operating system, including maintaining separate reference counts for each loadable version of the given software module; loading a replacement software module for the given software module into the operating system, including registering version information of the replacement software module with an operating system kernel; receiving a reference for the given software module after the replacement software module is loaded into the operating system; and directing the reference for the given software module to the replacement software module when the reference count is greater than zero based on the version information." Claim 8, especially as amended, recites similar subject matter. Support for the amendments can be found in paragraphs

[0013]-[0018] of the originally filed specification. Thus, Ma et al. and APA do not teach all of the limitations of the independent claims.

These differences are significant because the older version of the software module can be retained in memory while it is still being used by certain processes of the operating system, while new references can be redirected to the newer version in response to requests for application functions that are made by the kernel in a conventional fashion, which could not be accommodated by using an object validity variable as taught by Ma et al.. Use of the *invariable* version information for redirecting the references also significantly produces a more robust solution than using an object validity variable as taught by Ma et al.

Accordingly, Applicants respectfully request the Examiner reconsider and withdraw the rejection of independent claims 1 and 8 under 35 U.S.C. § 103(a), along with rejection on these grounds of all claims dependent therefrom.

Claims 7, 11, and 14-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Ma et al. (U.S. Pat. No. 5,920,725) in view of APA Admitted Prior Art and Corbet (Porting Drivers to the 2.5 kernel). This rejection is respectfully traversed.

The Examiner relies on Ma et al. to teach that each software module maintains its own reference count. However, Ma et al. teach at col. 10, lines 40-56 that replaced objects must be marked invalid in order for new references to be made to the presumed valid replacement objects. But it is not conventional for a Kernel to direct requests for application functions in this way. Also, tracking object validity, a *variable* state, for each object is an inconvenient and error prone process. Therefore, it is significant that Ma et al. do not teach, suggest, or motivate maintaining a count for each loadable module as

well as for each version of a loadable module, so that when the replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module based on version information registered with the kernel.

The Examiner merely relies on APA to teach that some execution environments, like Linux, allow replacement of a loadable software module by first removing the module. However, Ma et al. in view of APA do not teach, suggest, or motivate maintaining a count for each loadable module as well as for each version of a loadable module, so that when the replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module based on version information registered with the kernel.

The Examiner relies on Corbet to teach managing a reference count to an object outside the object. However, Corbet does not teach, suggest, or motivate maintaining a count for each loadable module as well as for each version of a loadable module, so that when the replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module based on version information registered with the kernel.

Applicants' claimed invention is directed toward loading a replacement software module without first having to unload a previous version of the software module. In particular, Applicant's claimed invention is directed toward maintaining a count for each loadable module as well as for each version of a loadable module, so that when the

replacement software module is loaded into the system and registers its version information with the kernel, new references created to an older version of the module can be redirected to the newest version of the module. For example, independent claim 1, especially as amended, recites, "maintaining a reference count for a given software module loaded in the operating system, including maintaining separate reference counts for each loadable version of the given software module; loading a replacement software module for the given software module into the operating system, including registering version information of the replacement software module with an operating system kernel; receiving a reference for the given software module after the replacement software module is loaded into the operating system; and directing the reference for the given software module to the replacement software module when the reference count is greater than zero based on the version information." Claims 8 and 14, especially as amended, recite similar subject matter. Support for the amendments can be found in paragraphs [0013]-[0018] of the originally filed specification. Thus, Ma et al., APA, and Corbet do not teach all of the limitations of the independent claims.

These differences are significant because the older version of the software module can be retained in memory while it is still being used by certain processes of the operating system, while new references can be redirected to the newer version in response to requests for application functions that are made by the kernel in a conventional fashion, which could not be accommodated by using an object validity variable as taught by Ma et al.. Use of the *invariable* version information for redirecting the references also significantly produces a more robust solution than using an object validity variable as taught by Ma et al.

Accordingly, Applicants respectfully request the Examiner reconsider and withdraw the rejection of independent claim 14 under 35 U.S.C. § 103(a), along with rejection on these grounds of all claims dependent therefrom. Applicants further request the Examiner reconsider and withdraw the rejection of claims 7 and 11 under 35 U.S.C. § 103(a) in view of their dependence from allowable base claims 1 and 8.

CONCLUSION

It is believed that all of the stated grounds of rejection have been properly traversed, accommodated, or rendered moot. Applicant therefore respectfully requests that the Examiner reconsider and withdraw all presently outstanding rejections. It is believed that a full and complete response has been made to the outstanding Office Action and the present application is in condition for allowance. Thus, prompt and favorable consideration of this amendment is respectfully requested. If the Examiner believes that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at (248) 641-1600.

Respectfully submitted,

Dated: July 24, 2007

By: Gregory Stobbs
Gregory A. Stobbs, Reg. No. 28,764

HARNESS, DICKEY & PIERCE, P.L.C.
P.O. Box 828
Bloomfield Hills, Michigan 48303
(248) 641-1600

[GAS/JSB]